EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

2. Authorization for this examiner's amendment was given in a telephone interview with James K. Okamoto on 05/08/2008.

3. The following claims had been amended:

1. A method of reducing access latency to a task priority register of a local programmable interrupt controller unit within a microprocessor, the method comprising:

receiving a command to write an interrupt mask value to the task priority register;

writing the interrupt mask value to the task priority register; and

writing the interrupt mask value into a shadow copy of the task priority register, wherein the shadow copy is written at a memory location each time that the task priority register is written;

receiving a command to read the interrupt mask value from the task priority register; and

reading the interrupt mask value from the shadow copy, instead of from the task priority register,

wherein the method obviates a need to use a serialize instruction after the task priority register is written because each interrupt performs a serialize operation and performs a read of an interrupt vector register to validate a current interrupt masking level.

2. (canceled).

3. The method of claim 1, wherein the shadow copy is always written after the task priority register is written.

4. The method of claim 3, further comprising, upon receiving an interrupt, reading the interrupt mask value from the task priority register and writing the interrupt mask value to the shadow copy.

5. The method of claim 1, wherein, if the task priority register is accessed frequently, then the shadow copy is stored in low-latency cache memory within the microprocessor.

6. (canceled).

7. The method of claim 1, whereby a latency of writing to the task priority register is substantially reduced.

8. The method of claim 1, whereby a latency of reading from the task priority register is substantially reduced.

9. The method of claim 1, wherein data in the task priority register reflects a level of priority of tasks being performed by the microprocessor.

10. The method of claim 9, wherein the task priority register comprises eight bits to designate up to 256 priority states.

11. The method of claim 1, wherein the method is performed by an operating system.

12. The method of claim 4, comprising, after writing the interrupt mask to the shadow copy, reading an interrupt vector register at a beginning of an interrupt handler.

13. The method of claim 12, further comprising executing instructions specific to the interrupt handler and returning from the interrupt handler.

14. A computer-readable medium storing an operating system with reduced access latency to a task priority register of a local programmable interrupt controller unit within a microprocessor, the computer-readable medium comprising:

microprocessor-executable code configured to write a priority level to the task priority register; and

microprocessor-executable code configured to write the priority level into a shadow copy of the task priority register, wherein the shadow copy is written at a memory location each time that the task priority register is written; and

microprocessor-executable code configured to read the priority level from the shadow copy, instead of from the task priority register, wherein the operating system avoids a need to use a serialize instruction after the task priority register is written because each interrupt performs a serialize operation and performs a read of an interrupt vector register to validate a current interrupt masking level.

15. (canceled).

16. The computer-readable medium of claim 14, wherein the shadow copy is always written after the task priority register is written.

17. The computer-readable medium of claim 16, further comprising, microprocessor-executable code configured to, upon receipt of an interrupt, read the priority level from the task priority register and write the priority level to the shadow copy.

18. The computer-readable medium of claim 14, wherein, if the task priority register is accessed frequently, then the shadow copy is stored in low-latency cache memory within the microprocessor.

19. (canceled).

20. The computer-readable medium of claim 14, whereby a latency of writing to the task priority register is substantially reduced.

21. The computer-readable medium of claim 14, whereby a latency of reading from the task priority register is substantially reduced.

22. (canceled).

23. (canceled).

24. (canceled).

25. (canceled).

26. A method of reducing a latency to read a task priority register of a microprocessor, the method comprising:

receiving a command to read an interrupt mask value from the task priority register; reading the interrupt mask value from a shadow copy at a memory location, instead of from the task priority register itself;

receiving a command to write an interrupt mask value to the task priority register;

writing the interrupt mask value to the task priority register; and

writing the interrupt mask value into the shadow copy of the task priority register,

wherein the method obviates a need to use a serialize instruction after the task priority register is written because each interrupt performs a serialize operation and performs a read of an interrupt vector register to validate a current interrupt masking level.

27. An operating system stored on a computer-readable medium with reduced latency to read a task priority register of a local programmable interrupt controller unit within a microprocessor, the operating system comprising microprocessor-executable code configured to:

write the interrupt mask value into the task priority register and into a shadow copy of the task priority register; and

read the interrupt mask value from the shadow copy at a memory location, instead of from the task priority register itself,

wherein a need to use a serialize instruction after writing the task priority register is obviated because each interrupt performs a serialize operation and performs a read of an interrupt vector register to validate a current interrupt masking level.

28. A multiple-processor computer system comprising:

a plurality of microprocessors interconnected by a processor bus, wherein each microprocessor includes a task priority register (TPR) with a interrupt mask value for that microprocessor;

a memory system, including local cache memory on each microprocessor and a main memory, wherein the memory system holds data including an operating system and shadow copies of the TPRs,

wherein the operating system includes executable-code for writing the interrupt mask value into the task priority register and into the shadow copies of the TPRs, for reading the interrupt mask values from the shadow copies and for maintaining the shadow copies, and

wherein a need to use a serialize instruction after writing the task priority register is obviated because each interrupt performs a serialize operation and performs a read of an interrupt vector register to validate a current interrupt masking level.

29. A method of reducing latency to write a task priority register within a microprocessor, the method comprising:

upon receiving a command to write an interrupt mask value to the task priority register, writing the interrupt mask value to the task priority register and into a shadow copy of the task priority register at a memory location without performing a serialization directly thereafter;

upon receiving a command to read an interrupt mask value from the task priority register, reading the interrupt mask value from a shadow copy at a memory location, instead of from the task priority register itself; and

upon receiving an interrupt, performing the serialization and reading an interrupt vector register, wherein a spurious indicator is returned if the interrupt is maskable,

wherein a need to use a serialize instruction after writing the task priority register is obviated because each interrupt performs the serialization and performs the read of an interrupt vector register to validate a current interrupt masking level.

Conclusion

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Abdullah-Al Kawsar whose telephone number is 571-270-3169. The examiner can normally be reached on 7:30am to 5:00pm, EST.

5. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng Ai T. An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

6. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Abdullah-Al Kawsar/
Abdullah-Al Kawsar
Patent Examiner
ART Unit 2195

/Meng-Ai An/
Supervisory Patent Examiner, Art Unit 2195